COMMENTS

The enclosed is responsive to the Examiner's Office Action mailed on January 7, 2008. At the time the Examiner mailed the Office Action claims 1-46 were pending. By way of the present response the Applicant has: 1) canceled claims 1-46; and, 2) added new claims 47-61. The Applicant respectfully requests reconsideration of the present application and the allowance of claims 47-61.

The Applicant has canceled claims 1-46 and has added new claims 47-61 rendering moot the basis for the Examiner's rejection. The Applicant, however, points out further below pertinent distinctions between new independent claims 47, 52 and 57. New independent claims 47, 52 and 57 (emphasis added):

> modifying methods of a class file to include additional byte code instructions at said methods' respective entry and/or exit points;
>
> modifying said class file to include a registration method information structure;
>
> registering said class file with a dispatcher, said registering occurring after said additional byte code instructions and said registration method information structure have been added to said class file, said registering of said class file including passing said class file's name, said methods' names and said methods' argument types to said dispatcher from said registration method information structure;
>
> said dispatcher performing the following in response to said registering of said class file:
>
> identifying a user defined plug-in pattern for said class file that lists one or more plug-ins for each one of said class-file's methods;

updating a dictionary maintained by said dispatcher with said plug-in pattern;

passing an identifier to said class file that uniquely identifies said class file;

executing said class file's methods in a run-time environment, said methods invoking said dispatcher upon reaching said entry and/or exit points, each invocation passing to said dispatcher: a) said identifier, b) an identity of the invoking method, and, c) the invoking method's arguments;

said dispatcher, for each invocation, using a), b) and c) above to look-up an appropriate plug-in from said dictionary and passing an identity of said appropriate plug-in to said invoking method so that said invoking method can receive appropriate handler treatment from said appropriate plug-in.

The above claim limitation recites at least the following points of distinction over U.S. Pub. Pat. No. 2004/0123279 (the Boykin reference):

1) a plug-in pattern and a dictionary;

2) a registration method information structure.

The Applicant respectfully submits that Boykin fails to disclose both a plug-in pattern "that lists one or more plug-ins for each one of [a] class file's methods" and a dictionary that is "[updated] with [the] plug-in pattern" and used to "look up an appropriate plug-in". The Boykin reference appears to disclose only a single data structure, a "registry",that could conceivably be used by the Examiner to meet the Applicant's claimed plug-in pattern or dictionary. Thus, whereas the Applicant's claims recite two structures (a plug-in pattern and a dictionary), Boykin merely discloses a single structure ("a registry"). See, Boykin, Fig. 2, para. [0036] and

[0039]. Moreover, the Applicant's claims recite that the dictionary is updated with the plug-in pattern. Again, because Boykin only discloses a single structure, Boykin fails to disclose that a plug-in pattern is used to update Boykin's registry and/or fails to disclose that Boykin's registry is used to update a dictionary.

Boykin also fails to disclose a registration information structure that is added to a class file and is used to pass the class file's name, the class file's method's names and the class file's argument types to a dispatcher. As the Applicant understands Boykin, a class file registers with a registry by passing only the class file's name (and not the class file's name, the class file's method's names and the class file's argument types). See, Boykin, para. [0037]. Moreover, the class file's name is not passed to a registry from a method information structure that was previously added to the class file. The Boykin reference seems to only state that hooks are added to a class file when the class file is loaded. During class loading, only the class name is passed (to the injector). The class name does not appear to be passed from a registration information structure that was previously added to the class file. See, Boykin, para. [0037].

CONCLUSION

If there are any additional charges, please charge Deposit Account No. 02-2666. If a telephone interview would in any way expedite the prosecution of this application, the Examiner is invited to contact Robert B. O'Rourke at (408) 720-8300.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP

Dated: /3-19-09/

/Robert B. O'Rourke/
Robert B. O'Rourke
Reg. No. 46,972

1279 Oakmead Parkway
Sunnyvale, CA 94085-4040
(408) 720-8300